

CS1353 - SOFTWARE ENGINEERING

SAMPLE SHORT QUESTIONS AND ANSWERS

UNIT-I

1.What is meant by Software Engineering?

The application of a systematic, disciplined, quantifiable approach to the development, operation & maintenance of software .(ie)The application of Engineering to Software.

2.What are the characteristics of Software?

- Software is engineered or developed, it is not manufactured in the classical sense.
- Software doesn't wear out.
- Although the industry is moving toward component based assembly, most software continues to be custom built.

3.What are the phases of Problem solving Loop?

Problem definition → identifies the problem to be solved.

Technical development → Solve the Problem through some Technology.

Solution Integration → Delivers the result

Status Quo → Represents the Current status affairs.

4.List out the activities of Linear Sequential Model.

Software requirement analysis → Understand the nature of the problem, information domain, function etc.

Design → Translates the requirement in to Software representation.

Code generation → Translates design in to Machine-readable form.

Testing → Test the Logical internals functional externals of the System.

5.Mention some of the drawbacks of RAD model.

- Not suitable for large-scale projects.
- Commitment of developers & customers are needed.
- Not appropriate when technical risks are high.

6.What are the types of changes encountered during the Support phase>

Correction – To uncover defects in the Software

Adaptation – To accommodate changes to its environment

Enhancement – To provide additional functionalities.

Prevention — To serve the needs of its end users.

7.Define Business process engineering.

When the context of the engineering work focuses on a business enterprise.

8.What are the elements of Computer based Systems?

Software: Computer programs, data structures, & related document

Hardware: Electronic devices, inter connectivity devices, Electro mechanical devices.

People: Users & Operators

Database :organized collection of information.

Documentation: Descriptive information

Procedures: Steps that define the specific use of each system element.

9. Define Software Lifecycle.

Software Lifecycle is the period of time beginning with the concept for a Software product ending whenever the Software is no longer available for use.

SLC = SLCM + Activities.

10. What are the functions of data architecture?

- It provides the information needed for a business function.
- Identify the data objects and their relationship.

Eg: Customer

Attributes: Company name

Contact information

Product information

Past purchase

11. Define System Modeling?

- Define the processes that serve the needs of the view under consideration.
- Represent the behavior of the processes and the assumptions on which the behavior is based.
- Explicitly define both
- exogenous & endogenous input to the model.
- Represent all linkages that will enable the engineer to better understand the view.

12. State the System Engineering Hierarchy?

The world view is composed of a set of domains (D_i), which can each be a system, or system of systems.

$WV = \{D_1, D_2, D_3, \dots, D_n\}$

Each domain is composed of specific elements (E_j).

$D_i = \{E_1, E_2, E_3, \dots, E_m\}$

Each element is implemented by specifying the technical components (C_k) that achieve the necessary function for an element.

$E_j = \{C_1, C_2, C_3, \dots, C_4\}$.

13. Mention some of the factors to be considered during System Modeling.

- Assumptions
- Simplifications
- Limitations
- Constraints
- Preferences.

14. What are the different architectures developed during BPE?

Data architecture → framework for the information needs of a business.

Application architecture → incorporates the role of people & business procedures.
Technology infrastructure → foundation for the data & application architecture.

15. Define Verification & Validation.

Verification: The set of activities that ensure that software correctly implements a specific function.

Validation: The set of activities that ensure that the software has been built is traceable to customer requirements.

UNIT-II

16. What is meant by System Requirements?

- Set out the system services and constraints in detail.
- Serves as a contract between the system buyer & the system developer.

17. What are the types of Software system requirements?

- **Functional requirements:** Services the system should provide.
- **Non-functional requirements:** Constraints on the services.
- **Domain requirements:** reflect characteristics of the domain.

18. Write down the functional requirement for an Library management system.

- The user should be able to search either all of the initial set of databases or select a subset of databases or select subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store.
- Every order shall be allocated a unique identifier.

19. Mention some of the Notations for requirements specification.

- Structured natural language: Use standard form or Templates.
- Design description language: Programming language is used.
- Graphical notation: Text annotations is used.
- Mathematical Specifications: Based on finite state machines or sets.

20. Write down the SRC for "Add a node"

Function	Add node
Description	Adds a node to an existing design.
Inputs	Node type, Node position, Design Identifier
Source	Node type, Node position → user Design identifier → database
Output	Design identifier
Precondition	The design is open & displayed on the user's screen
Side effects	None

21. Define Requirement Engineering.

Requirement Engineering is a process that involves all of the activities required to create and maintain a system requirements document.

The four generic Requirement Engineering activities are:
Feasibility study, Requirement Elicitation & Analysis, Requirement Specification, Validation.

22. Mention some of the process activities of Requirement Elicitation & analysis.

- Domain Understanding
- Requirement Collection
- Classification
- Conflict resolution
- Prioritisation
- Requirement Checking

23. What are the principle stages of VORD method?

Viewpoint identification: Identify the services provided to each viewpoint.

Viewpoint structuring: Grouping related viewpoints into hierarchy.

Viewpoint documentation: Refining the description

Viewpoint system mapping: Identify objects in OOD

24. What are the different types of checks carried out during Requirement Validation?

Validity checks

Consistency checks

Completeness checks

Realism checks

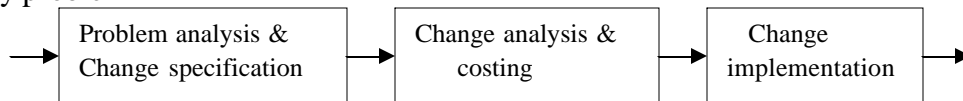
Verifiability.

25. Define Traceability

- Traceability is the overall property of requirements specification which reflects the ease of finding related requirements.
- Three types of traceability information to be maintained are:
 - Source traceability information
 - Requirement traceability information
 - Design traceability information

26. Draw the principle stages of Change management process.

Identify problem



27. State the primary objectives of analysis Model.

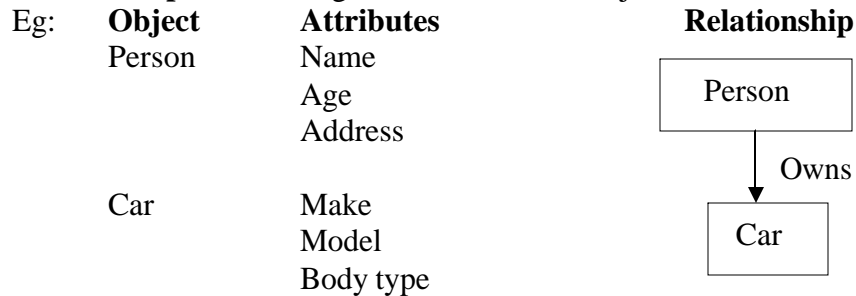
- To describe what the customer requires
- To establish a basis for the creation of a software design
- To define a set of requirements that can be validated once the software is built.

8. Define Data objects, attributes & relationship.

Data object: Representation of any composite information that must be understood by a software. It can be any external entity.

Attributes: Define the properties of the Data object

Relationship: Connecting two different data object.

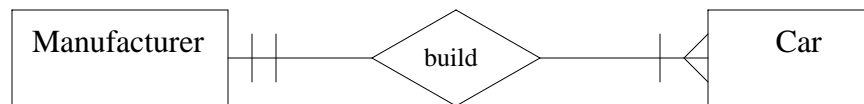


29. Define Cardinality & Modality.

Cardinality: Specification of the number of occurrences of one that can be related to number of occurrences of another.

Modality: The Modality of a relationship is 0 if there is no explicit need for the relationship to occur or the relationship is optional. The modality is 1 if an occurrence of the relationship is mandatory.

30. Define the Cardinal & Modality values for the data objects “manufacturer” and “Car”. The relationship is “builds”



31. State Entity /Relationship diagram.

ERD depicts relationships between data objects.

The ERD is the notation that is used to conduct the data modeling activity.

The primary components involved in the ERD are:

Data objects, attributes relationships and various types of indicators.

32. Define Data Flow Diagram.

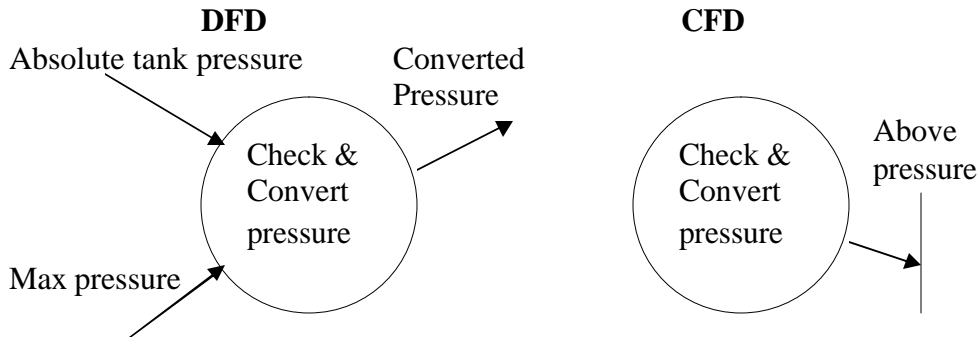
A Data flow diagram is a graphical representation that depicts information flow & the transforms that are applied as data move from input to output.

The basic form of a data flow diagram, also known as a data flow graph or a bubble chart.

33. What is meant by Information flow Continuity?

When we refine a fundamental model for a system, the information flow continuity must be maintained .ie, input & output to each refinement must remain the same.

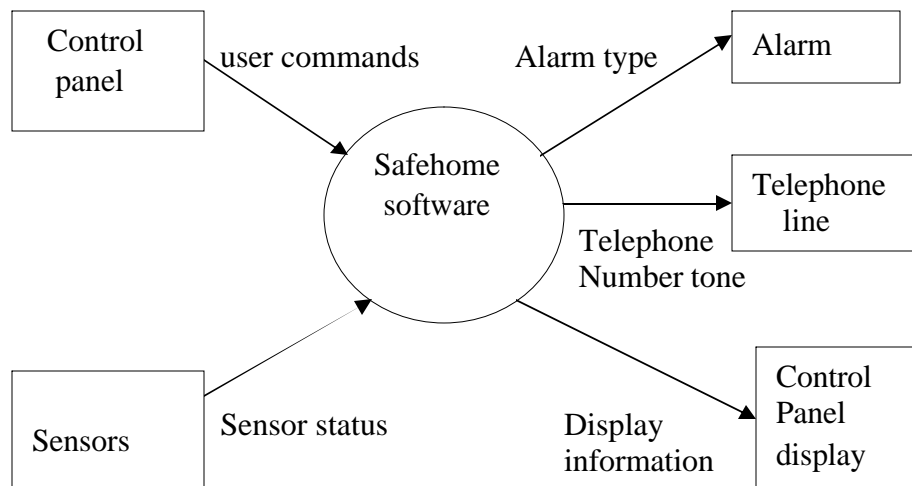
34. Draw a DFD & CFD of a test monitoring system for Gas Turbine engine.



35. Define Behavioral Modeling.

The state transition diagram represents the behavior of a system by depicting its states and the events that cause the system to change state.

36. Draw the Context level DFD for the Safe home Software.



37. What is meant by Data dictionary?

The Data dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions so that both user & system analyst will have a common understanding of inputs, outputs, components of store & intermediate calculations.

38. Define Process Specification.

Process Specification is used to describe all flow model processes that appear at the final level of refinement.

The content of the Process Specification include narrative text, a program design language, mathematical equations, tables, diagrams or charts.

39. What does data dictionary contains?

Name: The primary name of the data.

Alias: other names used

Where-used/How-used: A listing of processes that use the data or control item.

Content description: A notation for representing the content

Supplementary information: Other information like restrictions, limitations etc.

40. Write down the Data dictionary for the data item “Telephone Number”.

Names: Telephone number

Aliases: none

Where used/How used: assess against set-up

Description

Telephone number = [local number | long distance number]

Local number = prefix + access number

Long distance number = 1 + area code + local number

Area code = [800 | 888 | 561]

Prefix = * a three digit number that never starts with 0 or 1*

41. What is meant by Throw away Prototyping?

- Giving the user a system which is incomplete and then modifying and augmenting it as the user requirements become clear.
- The objective is to validate or derive the system requirements.
- Start with those requirements that are not well understood since you need to find more about them.

Unit-III

42. What is the use of Architectural design?

The Architectural design defines the relationship between major structural elements of the software, the “design patterns” that can be used, and the constraints that affect the way in which architectural design patterns can be applied.

43. Define Software design.

- Software design is an iterative process through requirements are translated into a “blue print” for constructing the software.
- The blue print depicts a holistic view of software.

44. Mention some of the Design principles.

- The design process should not suffer from tunnel vision.
- The design should not reinvent the wheel.
- The Design should exhibit uniformity & integration.
- Design is not coding, coding is not design.

- The design should be traceable to the analysis model.

45.State Procedural abstraction.

A Procedural abstraction is a named sequence of instructions that has a specific and limited function.

Eg: Open → Walk to the door, reach out & grasp the knob, turn knob & pull door
Step away from moving door.

46.What does Data abstraction contains?

A Data abstraction is a named collection of data that describes a data object

Eg: door Attributes: door type, Swing direction, Opening mechanism, weight.

47.What does Modularity concept mean?

Software architecture embodies modularity ;ie, Software is named in to addressable components called modules, that are integrated finally.

$$C(P1+P2) > C(P1) + C(P2)$$

The perceived complexity of a problem that combines p1 & p2 is greater than perceived complexity when each problem is considered separately.

48.Mention some of the criteria's used to define effective modular design .

- Modular decomposability
- Modular Composability
- Modular Understandability
- Modular Continuity.
- Modular protection

49.Define Fan-in & Fan-out.

Fan-out → A measure of the number of modules that are directly controlled by another module.

Fan-in → Indicates how many modules directly control a given module.

50.Differentiate horizontal partitioning & Vertical partitioning.

Horizontal partitioning
1.Defines separate branches of the Modular Hierarchy for each major program function
2.Propagation of fewer side-effects
3.Software is easier to maintain

Vertical partitioning
1.The Control & work should be distributed top-down in the program structure.
2.Higher probability of side effects.
3.susceptible to side effects when Changes are made.

51. Write down the concept of Functional independence.

- Functional independence is achieved by developing modules with “single minded” function and an “aversion” to excessive interaction with other modules.
- Independent modules ,is easier to develop because function may be compartmentalized,& interfaces are simplified.

- Independent modules are easier to maintain.

52.Distinguish between expected requirements and exciting requirements.

Expected requirements	Exciting requirements
Requirements are implicit to product Customer does not state them explicitly	Requirements reflect features that go beyond the customer's expectations and prove them to be very satisfying when present.

53.What is meant by software prototyping?

Software prototyping is a method of creating a model for the software product. This is developed based on the currently known requirements. Software prototyping helps in the better understanding of the required system. It is applied for complicated and large systems. In software prototyping, the customer defines a set of general objectives for the software, but does not identify the real input, processing and output requirements.

54.What is the work product of software design process and who does this?

A design model that encompasses architectural, interface, component level and their representations is the primary work product that is produced during software design.

Software engineers conduct each of the design tasks.

55.Define the term “software architecture.”

The software architecture of a program or a computing system is the structure of the system, which comprises software components, externally visible properties of those components and relationship among them. The architecture is the manner in which the various components of the building are integrated to form a cohesive whole.

Unit-IV

56.Define black box testing strategy.

Black box testing focuses on the functional requirements of the software. Test cases are decided on the basis of the requirements or specifications of the program and internals of program are not considered. Test cases are generated based on program code.

57.What is meant by software change?

Software change is defined as the change in nature of software as the requirements of software changes.

58.Why testing is important with respect to software?

A testing process focuses on logical internals of software ensuring that all statements have been tested and all are functional externals. While testing, we execute the entire program before it gets to the customer with specific intent of finding and removing all errors. In order to find the highest number of errors, test must be conducted systematically and test cases must be designed using disciplined techniques.

59. Write short notes on empirical estimation models.

Estimation model for computer software uses empirically derived formulas to predict effort as a function of line of codes (LOC) and function points (FP). The values of LOC and FP estimated are plugged into estimation model. The empirical data that support most estimation models are derived from a limited sample of projects. So, results obtained from models should be used judiciously. The model must be tested and compared with actual and predicted data

60. Justify the term “Software is engineered”

Software is engineered not manufactured. Although some similarities exist between software development and hardware manufacture, the two activities are fundamentally different. Both activities are dependent on people, but the relationship between people applied and work accomplished is entirely different. Both activities require the construction of a “Product” but the approaches are different’.

61. State Lehman’s Fifth law.

Over the lifetime of a system, the incremental change in each release is approximately constant.

62. Define software scope.

The first software project management activity is the determination of software scope. Scope is defined answering the following questions.

Context: What constraints are imposed as a result of the context.

Information objectives: What data objects are required for input?

Function and performance: Are any special performance characteristics to be addressed?

63. Define process maturity.

In recent years there has been a significant emphasis on process maturity. The Software Engineering Institute. (SEI) has developed a comprehensive model predicated on a set of software engineering capabilities that should be present as organizations reach different levels of process maturity. The grading schema determines compliance with a capability maturity model (CMM) that defines key activities required at different levels of process maturity.

64. Distinguish between alpha testing and beta testing.

Alpha test	Beta test
The alpha test is conducted at the developer site. The software is used in a natural setting with the developer “looking over the shoulder” of the user and recording	The beta test is conducted at one or more customer sites by the end user of the software. The beta test is a Live application of the software in an environment that

problems.	cannot be controlled by the developer.
-----------	--

65.What is software architecture.

The software architecture of a program or computing system is the structure or structures of the system, which comprises software components, the externally visible properties of those components and the relationships among them.

66.What is meant by software change?

Once software is put into use, new requirement emerge and existing requirements change as the business running that software changes parts of the software may have to be modified to correct errors that are found in operation, improve its performance or other non functional characteristics. This entire means that ,after delivery ,software system always evolve in response to demands for changes.

UNIT-V

67. Write short notes on estimation models.

Empirical estimation models can be used to complement decomposition techniques and offer a potentially valuable approach in their own right. A model is based on experience and takes the form

$$D=f(v_i)$$

Where d –number of estimated values

v_i –selected independent parameters

68.what is software measure?

A software measure is a mapping from a set of objects in the software engineering world information a set of mathematical construct such as numbers or vectors of numbers.

69.What is software metric?

A software metric is a simple quantitative measure derivable from any attribute of the software life cycle

(E.g.) LOC (Line Of Code)

Function point.

70.Define software measurement.

A software measurement is a technique or method that applies software measures to a class of software engineering objects to achieve a predefined goal.

71.What are the characteristics of software measurement?

- Object of measurement
- Purpose measurement
- Source of measure
- Measured property

➤ Context of measurement.

72.What is regression testing?

Regression testing is the re-execution of some subject texts that have already been conducted to ensure that changes have not propagated unintended side effects.

73.What is smoke testing?

Smoke testing is an integration approach that is commonly used when “shrink-wrapped” software products are being developed.

74.What is recovery testing?

Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.

75.What is security testing?

Security testing attempts to verify that protection mechanism built into a system will protect it from improper penetration.

76.What is stress testing?

A stress testing executes a system in a manner that demands resources in abnormal quantity, frequency, or volume.

77.What is performance testing?

Performance testing is designed to test the run-time performance of software within the context of an integrated system.

78.What is cyclomatic complexity?

Cyclomatic complexity is given by McCabe. The general formula to compute cyclomatic complexity is

$$M = V(G) = E - N + 2P$$

Where

E-Number of edges

N-Number of nodes

P-Number of unconnected paths of the graph.

79.What is CASE?

CASE tool stands for Computer Aided Software Engineering. Computer Aided Software Engineering tools assists software engineering manages and practitioners in every activity associated with the software process.

80.What is software testing?

The testing process focuses on the logical internals of the software, ensuring that all statements have been tested, and on the functional externals, that is, defined input will produce actual results that agree with required results.

81.What is structural testing?

Structural testing is testing the internal logic of a code.

82.What is statement coverage?

Statement coverage is the testing to test every statement in the code has been executed at least once.

83.What is branch coverage?

In branch coverage testing each branch in the code is executed at least once.

84.What is path testing?

In path testing all possible logical paths in a program are tested.

85.What is COCOMO model?

COCOMO stands for COnstructive COst MOdel. It is used for cost

$$\text{Effort} = a * \text{KDLOC}^b$$

Where a and b are two parameters of the model whose specific values are selected upon the class of the software system.

86.What is function point?

Function point is the size of measure it is also used to estimate the software cost.

Function point (FP)=Function Count (FC) * Value adjustment factor (VAF).

$$\text{Function Count (FC)} = \sum \sum x_{ij} w_{ij}$$

Where x - Number of elements
 w - Weight

$$\text{Value adjustment factor (VAF)} = 0.65 + 0.01 \sum f_i$$

87.What are business process engineering tools?

By modeling the strategic information requirements of an organization, business process engineering tools provide a “meta-model” from which specific information systems are derived.

89.What is a process modeling and management tool?

Process modeling and management tools are used to represent the key elements of a process so it can be better understood. Such tools can provide links to process descriptions that help those involved in the process to understand the work tasks that are required to perform it.

90.What is bottom-up approach?

System developed starting from detailed modules. Testing starts from the detailed modules and proceeds up to the higher levels of hierarchy.

91.Define top-down testing.

System developed starting from most general modules. Testing starts from the most general module.

92.Define big bang testing.

All modules integrated in a single step and tested as an entire system

93. Define sandwich testing.

Testing combines the ideas of bottom-up and top-down testing by defining a certain target layer in the hierarchy of the module. The modules below this layer is tested following bottom-up approach, whereas those above the target layer are subjected to top-down testing.

94.What is Parkinson's Law?

Parkinson's Law states that work expands to fill the time available. The cost is determined by available resources rather than by objective assessment.

95.What is software maintenance?

Changes to the software are made in response to changed requirements but the fundamental structure of the software remains stable.

96.What is architectural transformation?

This is a more radical approach to software change than maintenance as it involves making significant changes to the architecture of the software system.

97.What is software re-engineering?

No functionality is added to the system. Rather, the system is modified to make it easier to understand and change. System re-engineering may involve some structural modifications but does not usually involve major change.

98.Define Brooks law.

Software was delivered late, was unreliable, cost several times the original estimates and often exhibited poor performance characteristics.

99.What is Delphi model?

In Delphi method, a number of parameters to be determined based on as experts estimates.

The group estimate is taken as an average of the weighted individual estimates, computed as

$$\text{Estimate} = (\text{lower bound of estimate} + 4 * \text{most likely estimate} + \text{upper bound of estimate}) / 6$$

Variance=(upper bound of estimate-lower bound of estimate)/6

100.State Lehman's First law.

A program that is used in real world environment necessarily must change or become progressively less useful in that environment.

CS1353 -SOFTWARE ENGINEERING

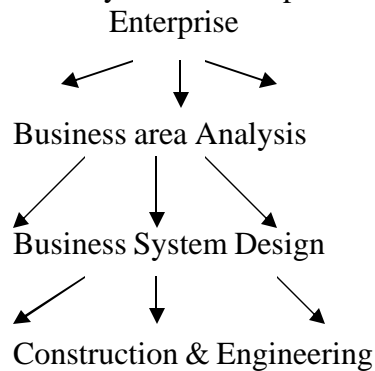
SAMPLE BIG QUESTIONS AND HINTS

1.Define Software process model? Explain any one of it with a neat diagram

The strategy that encompasses process, methods, and tools.

Explanation of any of the eight process model.

2.Explain the hierarchy of Business process Engineering .



3.Explain Software Life cycle process

Definition of SLC,SLCM

Predevelopment process

Development process.

4.Explain Evolutionary process model

Incremental model

WIN-WIN spiral model

Spiral model

Concurrent development model

5.Explain the different layers of Software Engineering?

Tools

Methods

Process
A quality focus

Unit-II

6. Build the following system:

“A Network based course registration system for your University”

Develop an ERD & a context level model for the above.

Create an ERD

Create a level 0 DFD.

7. Explain Transform Mapping with Safehome Software

Definition of Transform Mapping

Explain Safehome applications

Implement Transform mapping

8. Explain briefly Functional Modeling

Definition of Functional Modeling

Data Flow Diagrams

Extensions for real time systems

9. Explain the various Design concepts in detail.

Abstraction

Refinement

Modularity

Control Hierarchy

Structural Partitioning

Data structure

Information hiding

10. Discuss the various phases of Analysis Modeling.

Data Modeling

Functional Modeling & Information flow

Behavioral Modeling

The Data dictionary

UNIT-III

11. Discuss briefly Effective Modular Design.

Functional Independence.

Cohesion

Coupling

12. Explain User Interface Design activities

Interface design models

Interface design process

Defining Interface objects & actions

- Design Issues.
13. Discuss in detail Software configuration Management?
- The SCM process
 - Version Control
 - Change control
 - SCM standards
14. Explain Real Time Systems.
- Real Time Software design'
 - System Design
 - Real time executives
15. What is Software Architecture? Explain it.
- Definition
 - Explanation

Unit-IV

16. Explain in detail Black box testing in detail
- Syntax driven testing
 - Decision table based testing
 - Cause effects graph in Functional testing
17. How Boundary test conditions are achieved?
- Figure
 - Explanation
18. Explain in detail Structural testing?
- Statement coverage
 - Branch coverage
 - Path coverage
 - Example
19. Explain in detail Software Testing Strategies.
- Unit testing
 - Integration testing
 - Validation testing
 - System testing
20. Explain the test coverage criteria based on Data flow mechanisms.
- Basic block
 - All-use
 - c-use
 - p-use
 - du-path

Unit-V

21.Explain the various measures of Software.

- Program length
- Program volume measure]
- Potential volume measure
- Program level
- Effort & Time measure

22.Define Software Cyclomatic Complexity? How it can be calculated?

- Definition
- Figure & Explanation
- Formula
- Example

23.Explain how Software cost estimation can be achieved using Function point model?

- Figure & Explanation

24.Explain the COCOMO model in detail.

- The basic form of the COCOMO model.
- Explanation

25.Explain Delphi method of cost estimation

- Explanation with example

9. Draw the Schematic representation of Analysis activity with the Concurrent development Model.

